Organização e Arquitetura de Computadores II

Grupo F - Módulo de Previsão de Desvio

Bruno Brandão Inácio - 9838122

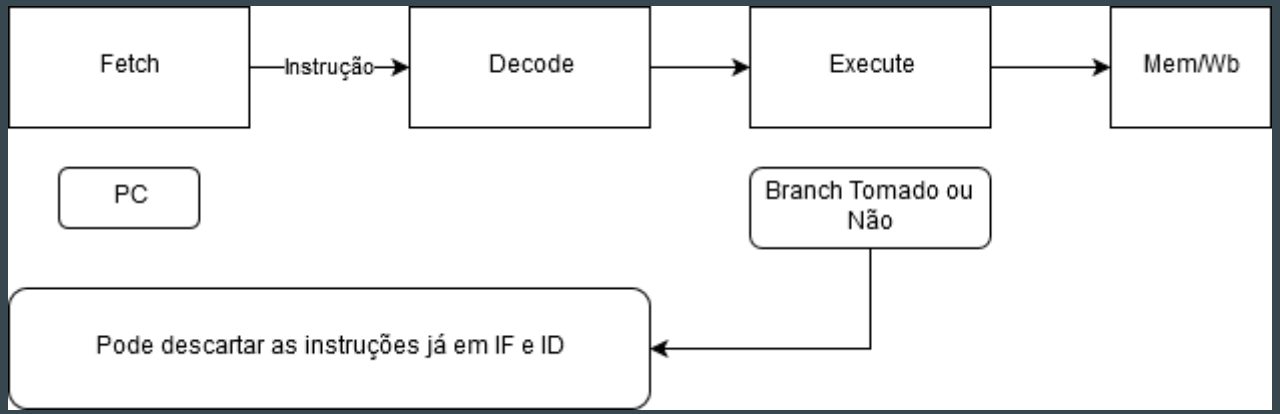
Pedro de Moraes Ligabue - 9837434

Pedro Henrique L. F. de Mendonça - 8039011

Vitor Tiveron de Almeida Santos - 9868085

Rodrigo Perrucci Macharelli - 9348877

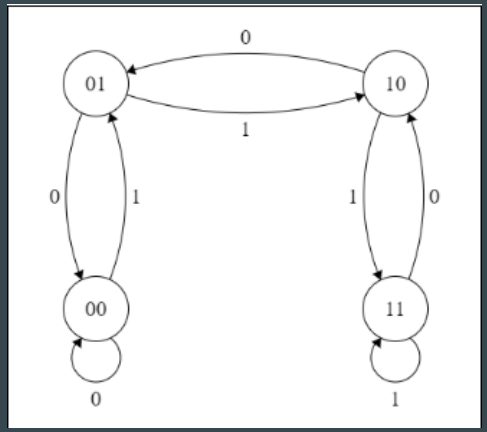
Por que previsão de desvio?



Previsor de Desvio

- Previsão é feita a partir de uma máquina de estado. Os estados definem qual será a previsão, conforme a tabela:

Estado	Descrição	Previsão
00	Fortemente não tomado	Sem desvio
01	Fracamente não tomado	Sem desvio
10	Fracamente tomado	Desvio
11	Fortemente tomado	Desvio



Previsor Global de Desvio

- Apenas mantém contagem dos desvios tomados ou não e toma decisões a partir dessa média.
- Analise o código abaixo supondo que o previsor está no estado 10.

```
int i = 0;
while (i<1000) {
        if (i < 900) {
                std::cout << "i < 900
        }
        ++i;
}
```

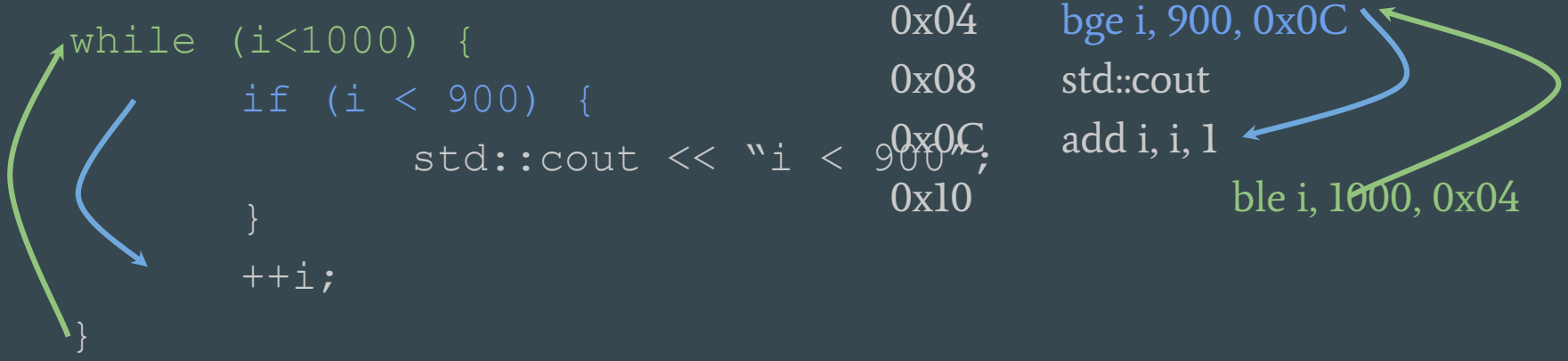
Estado	Descrição	Previsão
00	Fortemente não tomado	Sem desvio
01	Fracamente não tomado	Sem desvio
10	Fracamente tomado	Desvio
11	Fortemente tomado	Desvio

Previsor Global de Desvio

- Apenas mantém contagem dos desvios tomados ou não e toma decisões a partir dessa média.
- Analise o código abaixo supondo que o previsor está no estado 10.

```
int i = 0;
while (i<1000) {
        if (i < 900) {
                std::cout << "i < 900";
        }
        ++i;
}
```

```
0x00    mov 0, i
0x04    bge i, 900, 0x0C
0x08    std::cout
0x0C    add i, i, 1
0x10    ble i, 1000, 0x04
```

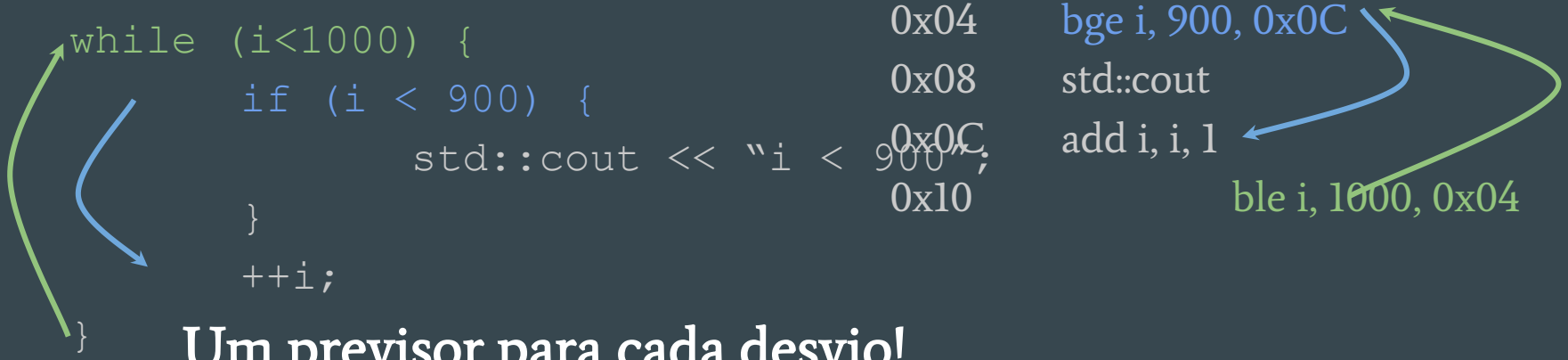


Previsor Global de Desvio

- Apenas mantém contagem dos desvios tomados ou não e toma decisões a partir dessa média.
- Analise o código abaixo supondo que o previsor está no estado 10.

```
int i = 0;
while (i<1000) {
        if (i < 900) {
                std::cout << "i < 900";
        }
        ++i;
}
```

```
0x00    mov 0, i
0x04    bge i, 900, 0x0C
0x08    std::cout
0x0C    add i, i, 1
0x10    ble i, 1000, 0x04
```



Um previsor para cada desvio!

Implementação - Tabela

- Em uma tabela, armazenam-se endereço da instrução atual, para que posição ela faz o branch e a previsão.

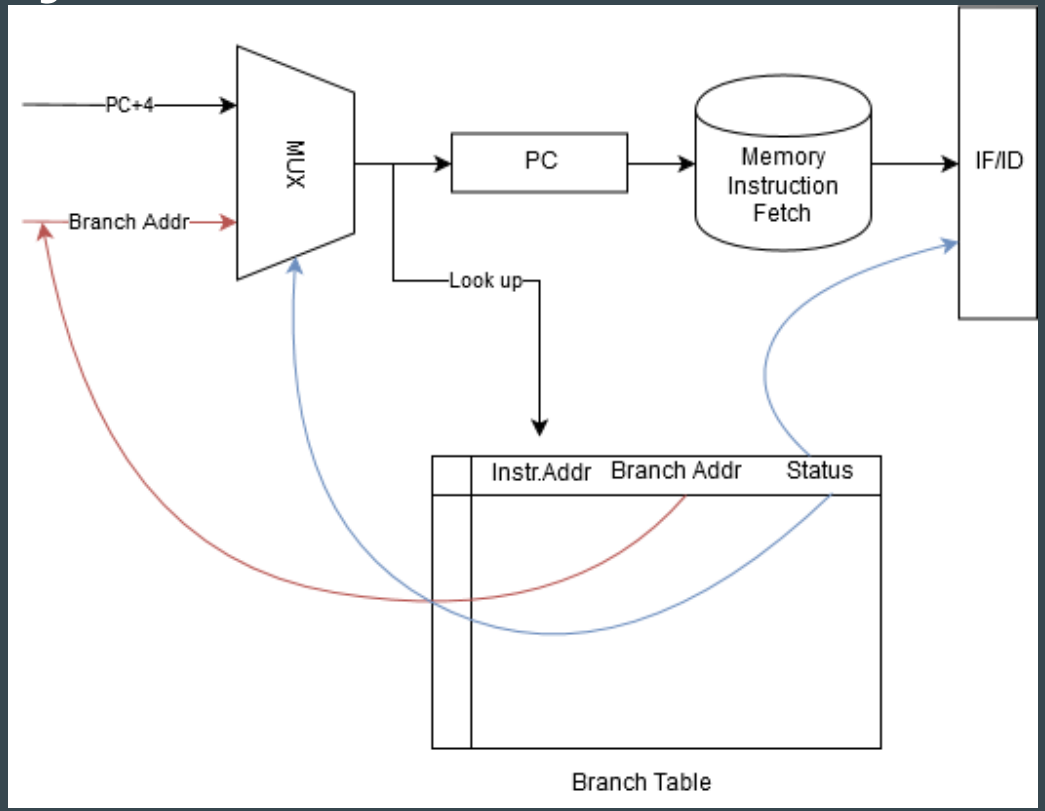
- Inserção por um ring buffer garantindo substituição FIFO

Instruction Addr	Branch Addr	Prediction
0x0012	0x0042	10
0x0016	0x0024	11
...		

Implementação - Interface da Tabela (Fetch)

```
17  entity branch_table is
18          generic
19          (
20                  addrSize        : NATURAL       := 8; -- tamanho do bus de enderecos
21                  tableSize       : NATURAL       := 16 -- quantos desvios guardar na tabela
22          );
23           port(
24           clock:    in   bit;
25           reset:   in    bit;
26
27     instruction_addrR: in bit_vector(addrSize-1 downto 0); -- end da inst atual
28     branch_addrR : out bit_vector(addrSize-1 downto 0); -- end para o qual desviar
29
30     prediction : out bit; -- se deve desviar ou nao
31
32  -- Esses sao usados no EXECUTE:
33                  branch_instruction: in bit; -- age como um enable de escrita
34                  branch_result : in bit; -- resultado do branch para atualizar
35                  instruction_addrW: in bit_vector(addrSize-1 downto 0); -- end da inst atual
36                  branch_addrW : in bit_vector(addrSize-1 downto 0) -- end para o qual desviar
37  -- se instruction_addrW nao existir na tabela, ele cria uma nova entrada, se nao, so atualiza.
38
39              );
40  end branch_table;
```

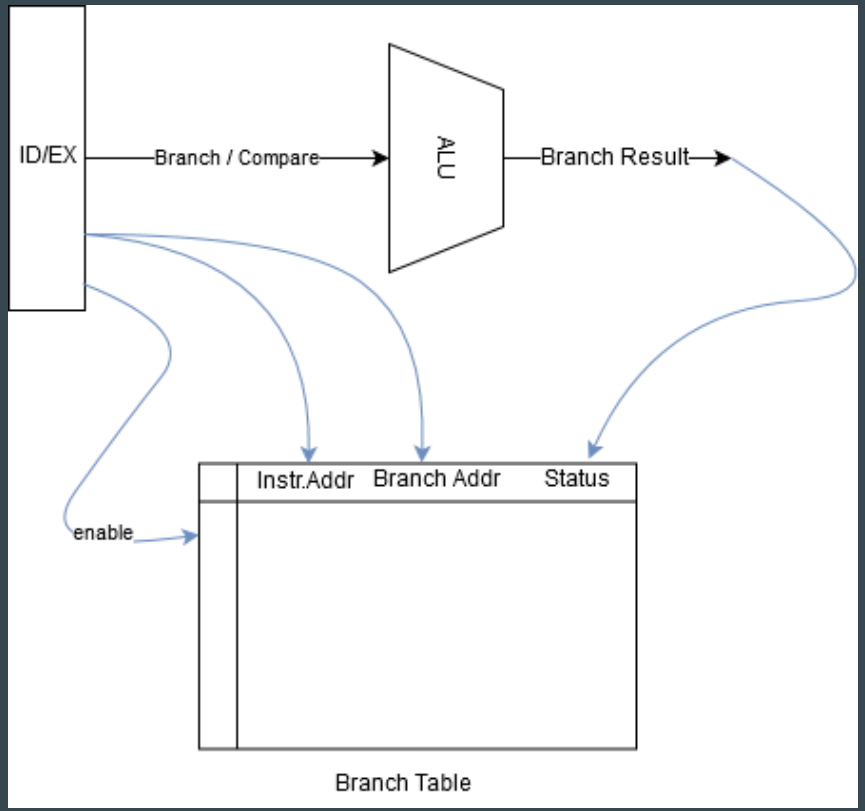
Implementação - IF



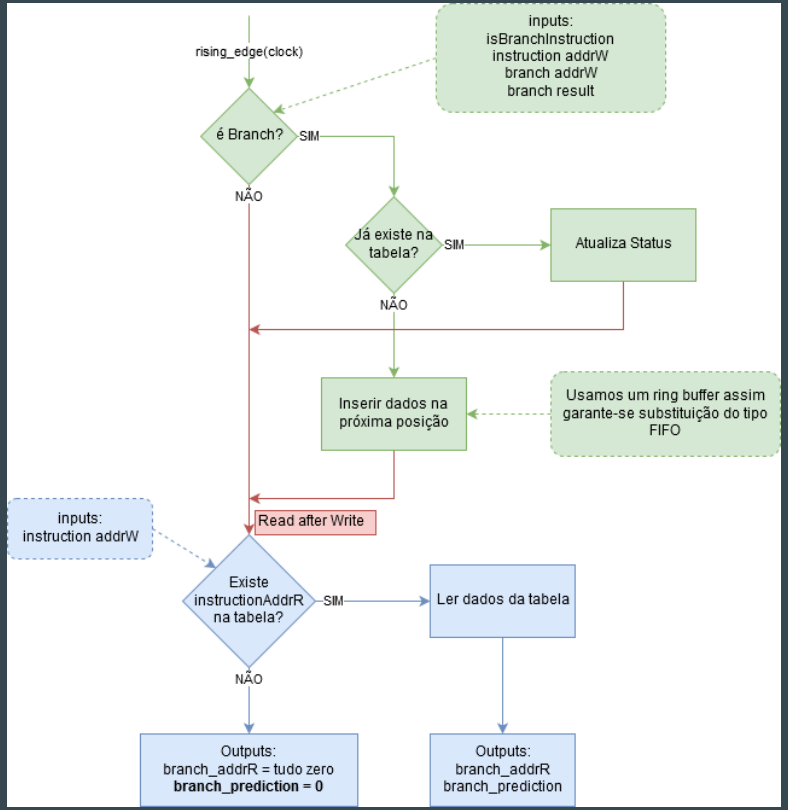
Implementação - Interface da Tabela (Execute)

```
17  entity branch_table is
18          generic
19          (
20                  addrSize        : NATURAL       := 8; -- tamanho do bus de enderecos
21                  tableSize       : NATURAL       := 16 -- quantos desvios guardar na tabela
22          );
23           port(
24           clock:    in   bit;
25           reset:   in    bit;
26
27  -- Esses sao usados no FETCH:
28                   instruction_addrR: in bit_vector(addrSize-1 downto 0); -- end da inst atual
29                   branch_addrR : out bit_vector(addrSize-1 downto 0); -- end para o qual desviar
30                   prediction : out bit; -- se deve desviar ou nao
31
32   branch_instruction: in bit; -- age como um enable de escrita
33
34   branch_result : in bit; -- resultado do branch para atualizar
35   instruction_addrW: in bit_vector(addrSize-1 downto 0); -- end da inst atual
36
37   branch_addrW : in bit_vector(addrSize-1 downto 0) -- end para o qual desviar
38
39              );
40  end branch_table;
```

Implementação - EX



Implementação - Descrição Funcional

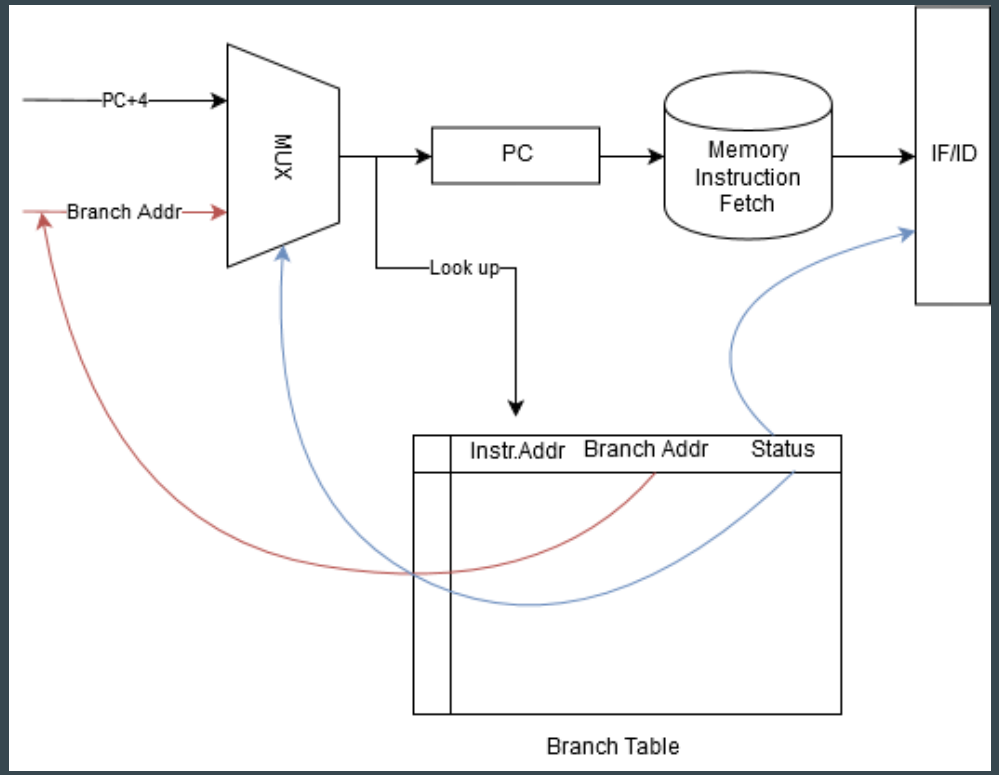


Implementação - Instruction Fetch

- PC_IN é um sinal de saída, usado como entrada na tabela de desvios;
- Novo multiplexador na entrada do PC, que recebe o endereço de desvio vindo da tabela;
- O seletor do multiplexador é equivalente à previsão da tabela, de forma que, se é previsto o desvio, o multiplexador seleciona a entrada com o endereço de desvio;

Dessa maneira, quando a tabela identifica uma instrução de desvio com probabilidade de ser tomado, o endereço seguinte carregado no PC já é o endereço de desvio. A previsão é armazenada no registrador de estágio.

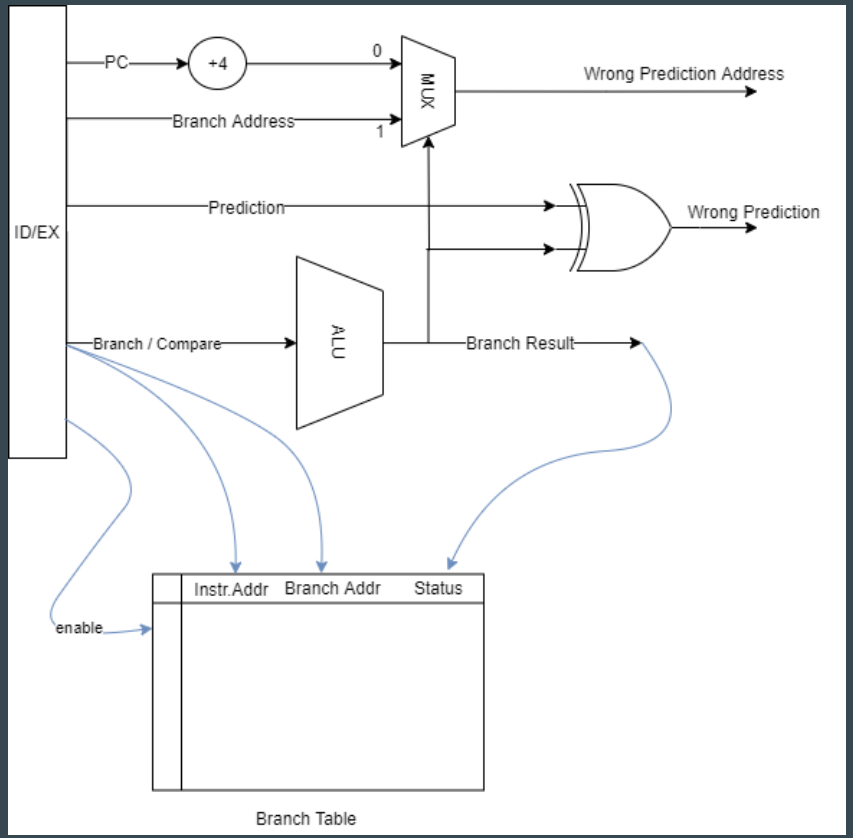
Implementação - Instruction Fetch



Implementação - Execute

- A lógica de desvio que estava no estágio Memory Access passa para o Execute;
- A lógica de desvio também é alterada, verificando se a previsão está certa ou errada, ao invés de se houve desvio ou não. Isso é feito usando um XOR com a previsão e o resultado do desvio.
- Novas saídas são criadas:
  - Endereço da instrução de desvio, endereço de desvio e resultado do desvio são enviados à tabela;
  - Sinal de descarte do pipeline são enviados aos estágios anteriores.
  - Para o Instruction Fetch, é enviado o PC do estágio + 4 ou o endereço de desvio, junto com um sinal indicando da previsão.

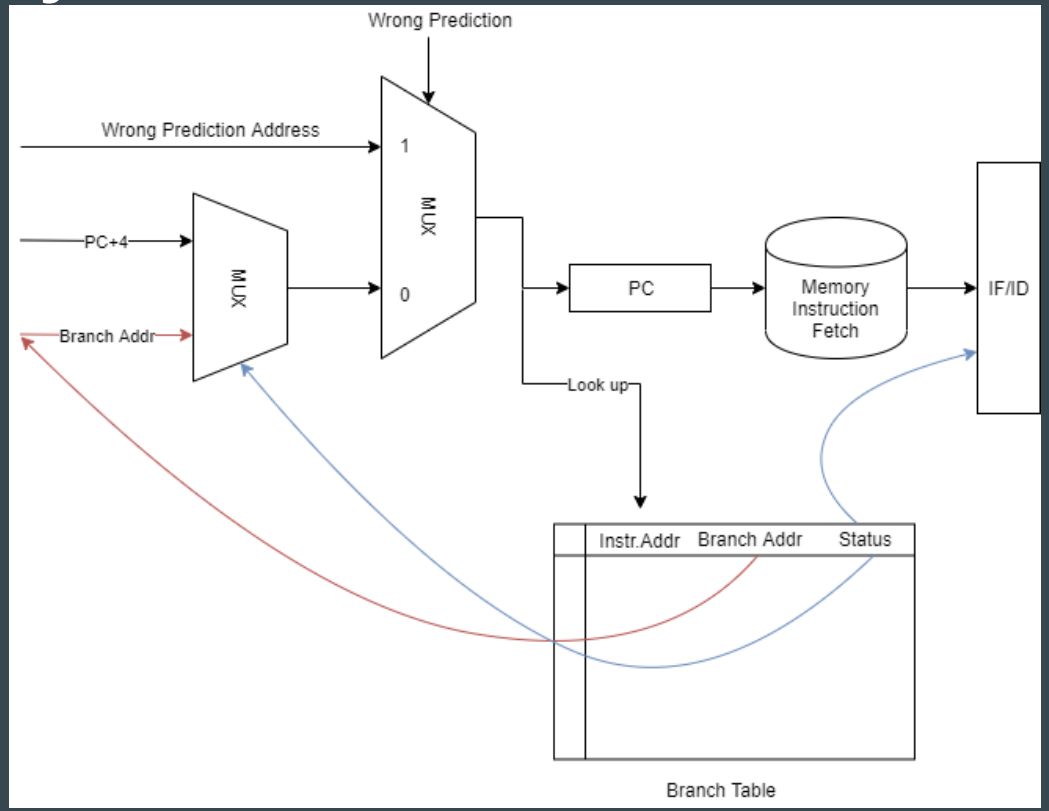
Implementação - Execute



Implementação - Conexões

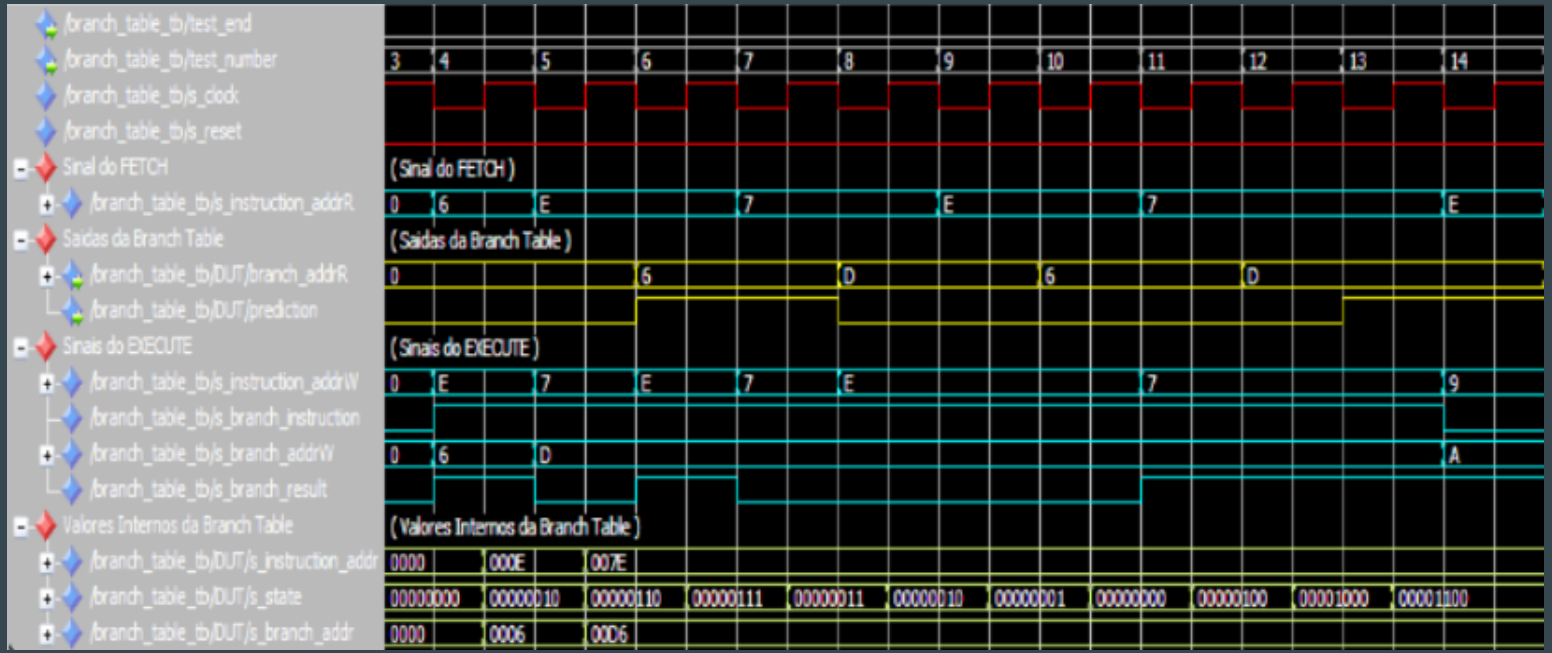
- Execute - Instruction Fetch
  - Em casos de erro de previsão, o Execute envia um endereço para o PC no Instruction Fetch, que pode ser tanto valor do PC do Execute + 4, caso não haja desvio, quanto o endereço de desvio, se houver desvio
  - O Execute também envia um sinal indicando o erro de previsão, de forma que o valor correto seja carregado no PC. Isso é feito utilizando um multiplexador imediatamente antes da entrada do PC.
- Descarte
  - O Execute envia o sinal de erro de previsão para multiplexadores imediatamente antes dos registradores de estágio. No caso do erro de previsão, os registradores carregam valores zerados, efetivamente descartando as instruções erradas.

Implementação - Conexões



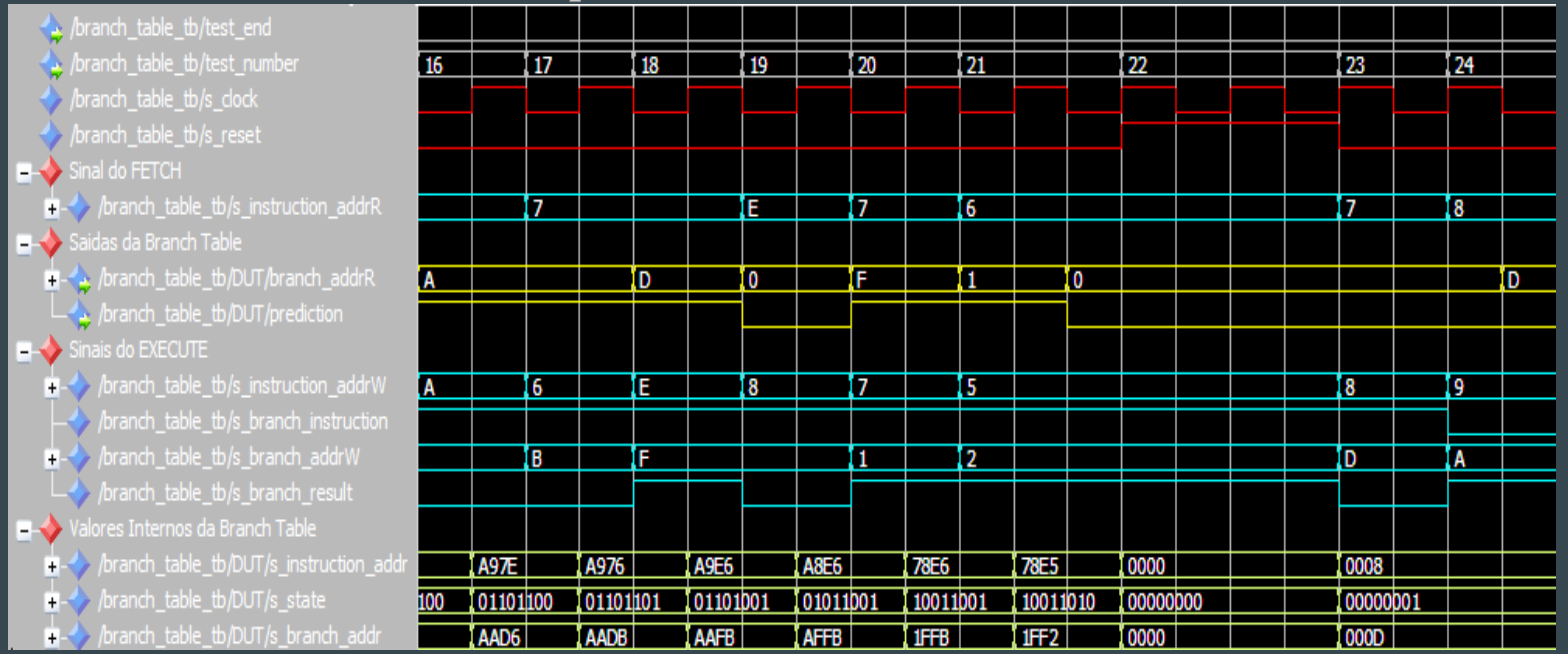
Testes

- Testes da tabela em funcionamento normal.



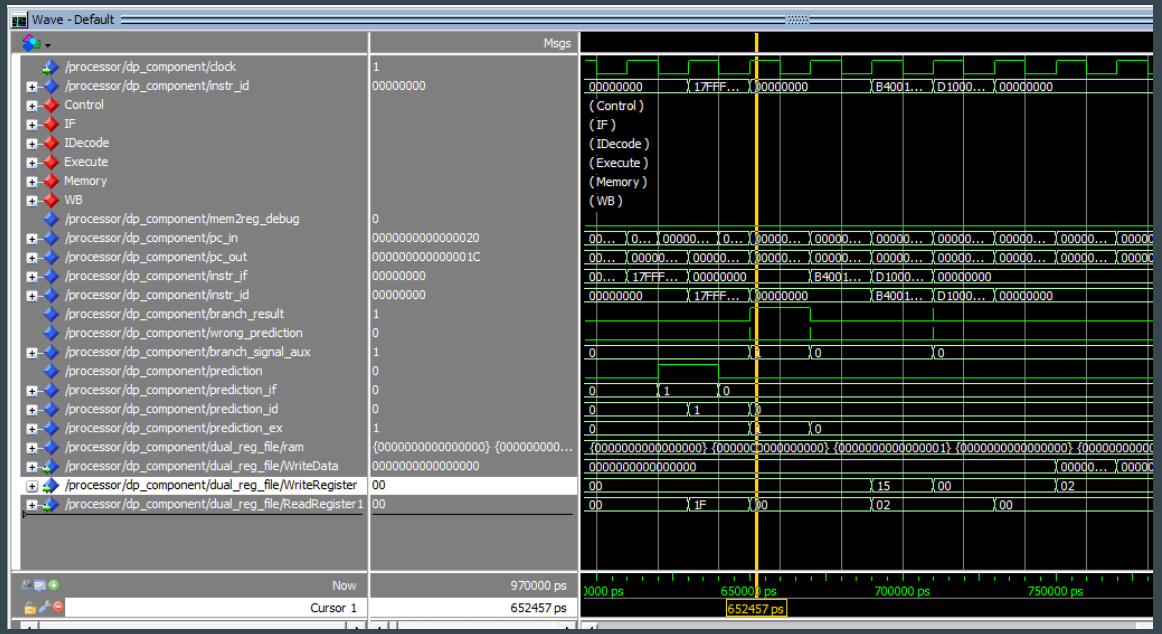
Testes

- Testes da tabela em casos especiais.



Testes

- Testes do pipeline realizados com o software ModelSim.



Perguntas?